Synchro-Phasor Data Conditioning and Validation Project

Phase 1, Task 3 Report

Algorithms and Methods for Data Validation and Conditioning

Prepared for the
Office of Electricity Delivery and Energy Reliability,
Transmission Reliability Program of the U.S. Department of Energy
Under Contract No.DE-AC03-76SF00098

Prepared by:

Electric Power Group

Ken Martin
Jianzhong Mo

Project Manager: Dejan J. Sobajic

March 21, 2014

Acknowledgement

The work described in this report was coordinated by the Consortium for Electric Reliability Technology Solutions, and funded by the Office of Electricity Delivery and Energy Reliability, Transmission Reliability Program of the U.S. Department of Energy through a contract with Electric Power Group, LLC administered by the Lawrence Berkeley National Laboratory.

TABLE OF CONTENTS

Figures

Synchrophasor Data Conditioning and Validation Project

Algorithms and Methods for Data Validation and Conditioning

1. Introduction

The Synchrophasor Data Conditioning and Validation Project sponsored by the US department of Energy Consortium for Electric Reliability Technology Solutions (CERTS) program was started in December 2012. The project objectives are to develop, prototype, and test various methods for conditioning and validating real-time synchrophasor data. The project is divided into three phases.

- Phase 1: Conceptual Design and Prototype Development
- Phase 2: Prototype Demonstration
- Phase 3: Functional Specifications of the Data Validation System

EPG approached the project by taking a broader look at data quality. Overall quality of data is affected by the equipment that takes the measurement, the communications used to gather it, and the equipment used to process and store it. Those in turn are affected by the overall design and management of the data system. With this in mind, the first task of Phase 1 was a survey of utilities that are implementing synchrophasor projects, particularly those sponsored by Smart Grid Investment Grant (SGIG). The purpose was to determine their data issues and how they dealt with them. It also provided information on the implementation and management of their systems to give some insights as to the source of successes and failures. The second task of Phase 1 was a best practices report based on what was observed in the interviews. Since the projects were typically not very far along, much of this second report was based on Electric Power Group, LLC (EPG) experience with the many projects they are involved in. The report focused on design, implementation, and administration of these systems.

The third task of Phase 1 is to research, design, develop, and test a prototype algorithm for validation and conditioning of the data itself. This is Report 3 of Phase 1 - Task 3 that covers the development of algorithm and the logic behind it.

2. Summary of Approach for Algorithms and Methods Used

The algorithm will be developed as a process that examines the data and makes modifications as well as creating a quality flag for each synchrophasor data variable. Validation assures that data is uncorrupted synchrophasor data. Invalid data is simply discarded and non-data filler is inserted to maintain the stream. Validated data is passed to a series of comparison modules which condition the data by flagging as valid, invalid, or uncertain. Invalid data values are set to non-numeric. Validated, conditioned data is combined with the flags in a C37.118 output stream. Applications use the validation and the conditioning flags for higher confidence in data use.

- Data Validation and Conditioning is divided into different sub-algorithms and processing modules. A data Quality Flag (QF) is created for each measurement to indicate when data has

been found to be good, bad, or uncertain. Validation – this is defined as checking the data in terms of frames, message length, CRC, time tags and other variables.

- Conditioning – this is defined as checking the measurements, first using data flags based on the C37.118 standard, next by self-checking based on high and low value limits/ranges, data noise and updated, and finally on matching with power system topology. Bad data is set to NaN and uncertain data is flagged.
- Data Output – Flags are used to characterize all data in terms of good, suspect or discarded. Users can decide which data elements to use based on the flags and use good data with the confidence that it has been validated and conditioned.

The data validation and conditioning is accomplished in 6 modules operating in a serial process, so that at each subsequent module, all of the flags are preserved. The first three modules perform data validation and modules 4, 5 and 6 perform data conditioning. The modules and the type of algorithms used in each module are listed below:

1. Module 1- Communication Interface: This module is designed to check for errors that may be introduced in the communications chain such as dropped bits, incorrect message frames, and CRC errors.
2. Module 2 – Message Characteristics: This module checks for message format errors such as length, destination address, type identification, and CRC16-check.
3. Module 3 – Timestamp: This module checks time tags for sequencing, data rates and transmission delays.
4. Module 4 – Quality Flags: This module utilizes all the flags available in the C37.118 standard to distinguish between good, bad, and uncertain measurements. Bad data is converted to NaN, suspect data is flagged, and all data is passed on to the next module for further processing.
5. Module 5 – Data Characteristics and Self-Checking: This module incorporates algorithms to check for unreasonably high or low values of voltage, current and frequency, data that is stale (not refreshing), and excessively noisy. Depending on severity, data that fails testing is declared bad and set to NaN or uncertain and flagged.
6. Module 6 – Topology Checking: The last module uses system topology to build algorithmic logic checking. For example, the sum of currents into a bus should be 0, and voltages at the same bus should be the same.

The algorithm process adds a data quality flag (QF) for each measurement as the data flows from one module to the next. The data output will be user selectable (or all choices to different outputs). One output will be the "cleaned" data in the same form as received. This will be the same data as received with the exception that all phasors will be in floating point polar, frequency and ROCOF in floating, the status for each will be updated according to problems detected by the algorithm, and the bad values set to NaN. All analogs and digitals will be included as received. This output will include the QF flags created by the algorithm. The second output will be the data as above except additional data can be converted to bad (NaN) as indicated by the user. The user can indicate which of the questionable and suspect data will be invalidated and which will be left to be used. This allows an application to determine from the data stream which data to discard without having to check a QF for each data item.

The project statement of work specified specific problems that the algorithm should be able to detect and flagged. The mapping of problems to the modules where they are addressed is provided in Appendix A to this report. For example, loss of data from one or several PMU's is addressed by module 2; loss of signals in a PMU, inconsistent data, and stale data are all addressed in module 5.

3. Considerations for Data Conditioning and Validation

A. Data Error Overview

This study deals with data as output from a Phasor Measurement Unit (PMU) (or Phasor Data Concentrator {PDC} after combination from several PMUs). The PMU may have very complex methods for computing phasor, frequency, rate of change of frequency, and other data values, but these are not the focus of this study. That is, PMU computation methods are not within scope of this study. It is assumed that the PMU meets the measurement requirements of C37.118.1 and has been correctly configured with naming and scaling. This study addresses only the data, the data timestamp, and the PMU quality flags that are specified in the C37.118.2-2011 standard. Other communication protocols may provide different data items and quality indications than those in C37.118. These are not discussed here; users will have to adapt their processing procedures according to the given definitions.

The EPG team first looked at the problem of data quality from the user perspective.

- What are their difficulties in using the data?
- What do they see that causes concern?
- What are they complaining most frequently about?
- What problems are being caused by bad data?

This assessment was largely accomplished for Phase 1 Task 2 for which best practices were derived. EPG found that the biggest part of the user confidence problem stems from the fact they see a lot of bad data (such as in plots) or the results of analysis using bad data, so naturally assume there is a lot of bad data and are unsure as to what they can trust. EPG also noted that the problems that cause bad data are often not repaired very quickly, so the bad data persists for longer than users expect it should. Bad data that continues for an extended time also diminishes confidence in the data. Given that synchrophasor systems are still evolving and not mature, they do experience higher percentage of bad data than other systems that they are intended to replace or supplement, so there is reason to investigate data issues.

Analysis of reported data problems showed that most resulted from the following sources:

1. **Communication** – this is the most common problem and usually results in loss of data. Problems range from inadequate bandwidth to failed equipment. Synchrophasor systems run continuously in real-time at high data rates and consequently are very sensitive to even short interruptions. Fairly complicated data exchanges seem to have problems with routing and firewall setups. While these may be "growing pains" associated with systems that are new to users, they do create a significant problem for extending the use of synchrophasor systems.

2. **Improperly installed and configured equipment** – this category of problem includes incorrect phasing, lack of compatible timing signals, and configuration errors. Hardware installations need to match the global use of the signal, so connection of the right phase to the right input is essential. The timing signal needs to include the required signal characteristics, as well as synchronization and quality indications as required by the PMU. Configurations not only set up reporting rates, but communication type details and algorithm characteristics. Since PMUs are in substations and generally require scheduling and travel to make changes, these problems often take significant time to resolve.
3. **Un-validated measurements** – once installed and operating, PMU measurements need to be validated against the signals they are measuring. This should be done at the substation to check the calibration and also done at the control centers to be sure the measurement is correctly mapped to the represented value.
4. **Inadequate problem monitoring** – problems can occur within the data in ways that are not immediately obvious. For example, loss of sync at a PMU will cause the phase angle to drift, but it may occur very slowly and not be noticed until it causes an alarm or other problems. Monitoring needs to include:
    a. Displays that allow the user to assess performance at a glance, including user settable thresholds for frequency of intermittent problems.
    b. Alarms that will alert users to problems, also at user settable limits.
    c. Continuous logging of problems for forensic analysis.
5. **Inadequate maintenance procedures** – system maintenance needs to be comprehensive with procedures that support quick and accurate problem resolution. Trouble maintenance—as opposed to routine maintenance—starts as soon as the system is operating. Synchrophasor systems include timing, power system signals, communication, processing, and display/analysis in an almost monolithic system. It is difficult to observe a signal or analytic result that looks "suspicious" and pinpoint the problem that caused it. Analytic components are needed to track and record information to determine the source of particular problems. Procedures are needed to use this information to go locate the problem source and effect repairs. Procedures also need to take into account the different operational groups involved, including the fact in some cases they will be in different companies. Most systems are not very old, so lack of routine maintenance is not a significant problem at this time.

Other problems have been observed such as measurement algorithm errors, but these are not common. This first effort focuses on detection of problem issues using methods that should be available to all users. Once these are addressed and it is shown that the approach can be successful, methods for more complex problem detection can be deployed.

B. Detecting and Tracking Data Errors Through The Measurement System

Data quality is affected by errors that can occur at any stage of processing, from sensing of the engineering quantity to be measured to display or analysis. Figure 1 illustrates a typical phasor measurement system. It starts with sensing of line voltages and currents, proceeds through measurement, communication, and completes with applications for display or analysis. Seven stages are indicated for analysis of error sources. Processing at each of these stages is detailed in the following sections.

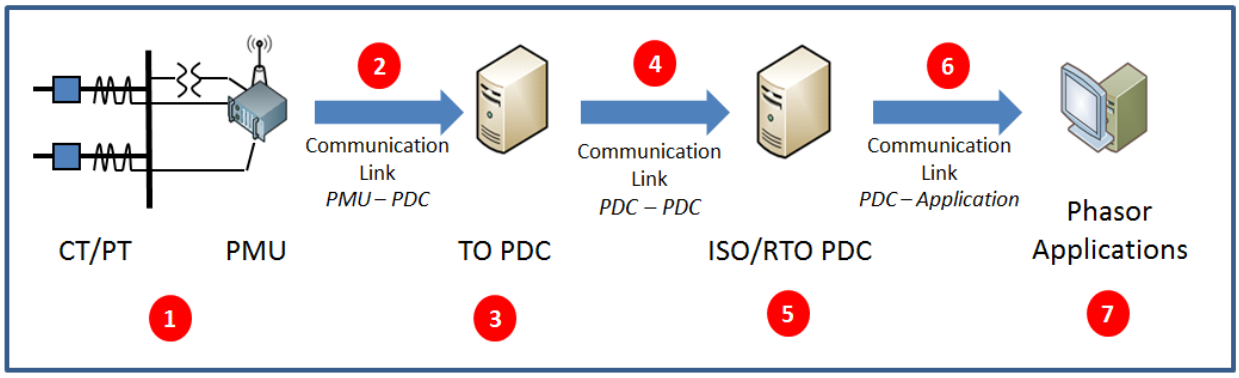

Figure 1. Phasor measurement system showing stages of processing

Stage 1 – Sensing and Measurement

Line voltage and current are sensed by potential transformer (PT) and current transformer (CT) transforming devices which could employ any of a number of transforming techniques. Different techniques exhibit different accuracy, bandwidth, and environmental stability characteristics. Since the PMU itself has high accuracy and uses a comparatively wide bandwidth, primary sensing and signal conduction (wiring) can affect phasor measurement. Basic characteristics of these devices are well documented and specified, but detailed analysis of all parameters that affect synchrophasor measurement remains to be studied. For the purposes of this study and report, the accuracy of the PT/CT is accepted as very high and the bandwidth sufficient to make phasor calculations equally accurate. Note that phase delay will cause errors in phase angle measurement and magnitude inaccuracy will reflect into the phasor magnitude.

PT/CT signal secondaries route directly to the PMU. The PMU uses shunts or instrument PT/CT devices to convert signal levels for the A/D input. They are considered to be part of the PMU whether external or internal. Once digitized, the PMU uses a numerical algorithm to estimate the phasor value. It uses the time input to align the estimation with universal time (UTC). Frequency and rate of change of frequency (ROCOF) are estimated from the AC inputs as well, usually from the phase angle of the phasor but other methods are used in some cases. Once estimated, the phasor and frequency measurements are combined in a packet (called a frame) with the estimate time-tagged and transmitted to a PDC or other collection device. The C37.118 data communication format includes a quality flag that indicates several types of measurement quality and error considerations that are summarized below:

1. If there is a fatal error in the phasor/frequency estimation processes that would cause the estimated values to be invalid, the PMU can set the invalid flag (bit 15). This one bit invalidates

all data reported in the frame and does not differentiate which reading is invalid, so is generally not used by the PMU. If reporting in floating point, the PMU could set the failed measurement to Not a Number (NaN) to indicate error and leave the rest valid. This option is rarely used at this time.

2. Since the process is numerical, the processor can self-check its components (both hardware and firmware) as well as the estimate itself. This can be indicated in the quality flag as a PMU error (bit 14). Bits 14 & 15 can be set together to indicate test mode.
3. Current timing devices have capability to determine whether they are in sync with a universal source. Some even can determine how closely they are in sync. The PMU indicates whether the measurement is synchronized with UTC (bit 13), how close the synchronization is to UTC (bits 6-8,) and how long synchronization has been lost after losing sync (bits 4-5).

Stage 2 – Communication from PMU to TO-PDC

Communication here refers to data transport from a substation to a PDC at a regional or central location. Data may first go from a PMU to a local substation PDC, but that function will be approximately the same as described in Stage 3 and is discussed below.

Communication systems and types will vary. Originally data was carried over on RS232 serial communication. That in turn likely went through analog modems and over some type of analog communication path. Now most communication is handled as a digital network using underlying communication adapted to digital data. The big difference is the type of error found and error frequency. Digital data converted to analog is prone to individual bit errors that are only detected by looking at the whole data package. The 16-bit CRC specified in the C37.118 standard is intended to detect these. Network communication tends to lose less data, but when it does, it usually loses full packets rather than just single bits. There is no error detection or monitoring at this stage.

Stage 3 – Transmission Operator (TO) PDC Processing

The PDC may incorporate a number of functions ranging from time alignment of data to format conversion and measurement adjustments. For this analysis, the PDC is assumed to only receive data, time align, and send the data. Much of the value of phasor measurement derives from the fact the measurement is precisely synchronized so represents a snapshot of the power system at an instant in time. Further, the angle measurements across the system can be compared only when they are made at the same instant of time. Consequently, having a PDC to align data from multiple sources has become an essential feature of phasor systems. The PDC has to include interfaces to receive data from multiple sources using available communication methods. It has to incorporate algorithms for time alignment under various scenarios. It has to be able to send the aligned data in aggregated packages.

A major consideration for a PDC is how to handle missing data. Data is sent continuously at a high rate, 10-60 frames/second. The PDC has a specified list of PMUs that is it receiving data from. It has to wait long enough to receive data with a specific timetag from all those sources so the measurements can be packaged together and sent on to the next stage. If a frame does not arrive or arrives with an error that makes it unusable, the packet needs to be sent along without the data. However, simplified data packaging most frequently employed (such as with C37.118) requires some kind of placeholder or “filler”

data to keep the packet interpretable. This filler can be flagged to indicate it is invalid, but that requires all successive processing to read and correctly interpret the flags.

The error and quality indications at this stage include:

1. The communication interface can detect some data errors and report them to the application program (in this case the PDC). These include framing errors (RS-232), synch word errors, CRC errors, packet size errors, and other communication related errors. However most communication interfaces simply discard the data by default when errors are detected, and this is mode programmers typically use. Consequently a communication interface error appears as a lost character or message. As noted above, missing messages require filler data and a flag to indicate it is invalid (bit 15). The standard C37.118-2011 requires filler data to use Not a Number (NaN) values to discourage incidental use further down the line.

2. C37.118 requires a CRC on every data frame. The PDC needs to check that this is correct. If not, the data in the frame has an unknown error and all the data should be marked as invalid. A framing error will cause a CRC error. An unannounced change in data content or format will cause a CRC error. The CRC is a catchall error detection method that may be used to trigger further investigation. In all cases, a frame with a CRC error should be discarded. In practice, systems that report error type counts show very few CRC errors, which indicates communications usually does not have bit and packet errors within a frame of data.

3. If the data frame is valid, it can be parsed to examine individual components. The first part to examine is the frame structure. The first byte must be a specific sync bit. The message length is indicated and the message should match. The ID for the data stream should match that expected. Most of these errors will cause a CRC error as well. However in the event they do not, they may be an indication of incorrect routing or data tampering.

4. The timestamp can be used to indicate a number of possible errors. Data is sent continuously without delay and in the order of measurement. If the data arrives out of order or in blocks with longer gaps between blocks or with much longer delays than are indicated there may be problems in the communication systems. It could also mean the data itself could be compromised, though this is unlikely. The timestamp accuracy and synchronization status is indicated by the C37.118 flags as described above. Time is what ties the phase angle measurements together; the time used to calculate phasor angles must be the same for comparisons between angles. Phase angles calculated using a common time source can always be used together. This may occur in a single PMU or in a substation where one clock supplies several PMUs even though the clock is not synchronized to UTC. However when measurements are made using separate clocks, the clocks must be synchronized to a common source. When the flags indicate the time quality is below an acceptable level, the phase angle should be marked as invalid. If the time stamp from a particular PMU is significantly in error, it may be impossible to synchronize its data with the other PMU's data by time stamp (e.g., the time stamp can be off by hours or days). In this case, the data is more accurately synchronized by assigning a timestamp that is the current PDC time and placing in the data package accordingly.

This "sort by arrival" time stamping works because the data is being sent in rea- time with minimal delay, so assigned time is close to the actual measurement time. (This is also the typical method timestamps are applied in SCADA.) This process only aligns the measurement amplitude; the phase angle was estimated with an invalid time and cannot be used, so the sort by arrival flag needs to be set (bit 12).

Stage 4 – Communication from TO PDC to ISO/RTO PDC

Communication at this level will almost always be high-bandwidth networks. Since data has been aggregated once, the volume for each link between these systems will be too high for serial or analog systems. This link may use utility or leased communications. The chief problems will be overloaded communications and firewall/security considerations. Administration needs to include documentation of links, routing, and other settings so that outages can be rapidly tracked down and traffic restored. Quality issues are chiefly the loss of entire streams. Some dropouts may occur if shared facilities are used and they result in occasional overloads.

Stage 5 – Independent System Operator (ISO) or Regional Transmission Operator (RTO) PDC Processing

This processing is essentially identical to that done in stage 3. The differences are that the PDC must have greater capacity for data and will receive several streams, each of which has a number of PMUs. A data frame may be divided into a number of packets for transmission, but the communication interface will reassemble the packets into a frame. The data block from each PMU has its own status indication. If a frame is dropped or the CRC is bad, all the individual status words need to be updated and the individual data blocks need to be filled with NaN.

The same time stamping issues apply as for the TO PDC. Ordering and latency provides indication of communication issues. If there is a timestamp error or the packet has to be sorted by arrival, all of the individual PMU status indications have to be updated.

Stage 6 – Communication from PDC to Applications

Phasor applications could be found at the TO or the ISO/RTO and possibly other places; this link is the same regardless of where it is found. It is the output of a PDC so will require higher bandwidth than a single PMU. This link will practically always be on a network. The same kind of considerations as in stage 4 will apply; the chief problems will be overloaded communications and firewall/security considerations. Administration needs to include documentation of links, routing, and other settings so that outages can be rapidly tracked down and traffic restored. Since applications are nearly always local to the PDC, dropouts and loss of link will be rare except where there are overloads and network configuration issues.

Stage 7 – Application Processing

Applications (as defined here) receive a single data stream that has already been processed by a PDC. No alignment, correlation, or error checking is needed, other that processing the data in order as indicated by the time stamp. Data that is lost may be approximated by the application or may be left

blank. The same processing considerations as described for the ISO/RTO PDC apply for dealing with replacing lost data.

Two categories or applications should be considered: those that simply store data for later use and those that use the data in real-time. The former may be called archiver. It may store in individual files, an historian, or a relational database. In all cases the storage system must keep up with the input data rate and there must be enough storage capacity for the intended data block size. There are no other data quality considerations. Quality flags are stored with the data and need to be processed by the analysis and other applications that use the data.

Applications that use data in real-time need to process the existing quality flags along with any further processing that is needed for the particular display, process, or alarm. Some additional processing is certainly advisable for many applications, but these are beyond the scope of this simple stage by stage analysis. These will be addressed in the next section.

C. Data Errors Not Tracked in the Measurement Processing Chain

The previous section described the C37.118 quality indicators as used and passed through the measurement chain. These are the most frequently encountered errors and therefore have assigned quality indicators. Other data quality issues will occur that need to be detected and processed. These data issues include but are not limited to:

1. Improper identification of signals.
2. Scale and offset errors in the measurements.
3. Inaccurate calibration.
4. Measurement limitations.
5. High noise on a signal.
6. Measurement values that repeat for a long period of time (stale data).
7. Undetected time synchronization errors.

Some of these problems can be detected by examination of the data itself, including some information about the data processing. These problems are referred to as data centric, since they only involve examining the data itself. Other problems can only be detected by using knowledge of the power system to which the measurements apply. These problems will be described as topology based since the topology of the power system is required for their detection.

Comprehensive installation procedures can identify most of these problems initially. Things like correctly identifying the signals and finding scale or angle offset errors can be located and resolved by comparison with SCADA and other measurements. Calibration can be checked in the lab as well as at installation. Installation procedures are described in Appendix A of the Best Practices Report. However, experience has shown that over time any one of these problems can occur as changes are made in the power and communication systems. Consequently continuous data monitoring is needed to alert users to errors that come up.

Data centric monitoring includes detection of stale data and noisy data. Stale data can be caused by a failed component that keeps outputting the same data, or by a component that repeats old data to act

as filler for missing data. Noisy data can be caused by high noise on a signal input (valid noise) or a failed component that produces bad signals. Both problems can be detected by looking at the data itself and applying identification techniques.

Extended data centric monitoring looks at data dependencies, such as frequency estimation that is derived from phasor calculation. If the input for the phasor fails, the frequency estimate will fail too. This monitoring requires some knowledge of the measurement equipment (PMU), though in most cases all equipment may follow the same pattern.

Some data problem detection requires knowledge of the system that the signals represent. This can be divided into simple measurement topology and comprehensive topology. For the simple topology case, sums and other simple comparisons can be used. For example, if measurements include all currents into a particular bus, they should sum to zero; if they don't (within an error tolerance), one or more of the measurements is inaccurate. This principle can be extended to multiple measurements of the same quantity, power flow, and other definable examples.

Comprehensive topology requires a system model to which the measurements can be applied. In this case, measurements will be compared with a complete representation of the system to which they apply. This technique allows validation of a number of measurements at once and of different types. It can be used to estimate quantities that are not directly measured as well. It can reveal calibration errors that are not exposed by other monitor techniques. It is probably the most reliable method for finding incorrectly identified signals and phase and synchronization problems that are not flagged. However like state estimation, it is a major investment in effort and introduces a new set of error sources in itself.

A very important point here is comparing the number of measurements with the number of variables to be determined. Traditional state estimation requires more measurements than variables in order to produce error indicators that show how good the estimate is. The same principle applies to phasor data in linear state estimation or its equivalent. Over-determined equations are required to show if there are measurement or calibration errors. If there are not enough measurements to produce some redundancy, it is not possible to detect errors. A phasor advantage is that the measurements are relatively independent. There could be enough measurements at a station or in an area to find measurement errors at that location, yet not in the whole system. The principle of over-determination applies but has a different scope than in traditional state estimation.

D. Selecting Algorithms for Data validation and Conditioning and Implementation of Algorithms

After evaluation of the requirements and approaches discussed in the preceding section, EPG decided to design and implement the validation algorithm along the following lines:

1. Keep the algorithm as simple as possible while detecting most problems (including those indicated in the Lawrence Berkeley National Lab RFP).
2. Use the existing C37.118 flags as the first line of data problem detection.
3. Add problem detection without increasing complexity.

4. Include options to imbed problem determination in the data itself.
5. Add a measurement quality flag for each measurement that can be carried in the data.
6. Allow the user to select data with flags to self-determine which to discard.
7. Allow the user to receive fully conditioned data where bad or uncertain data is set to NaN.

1. Keep it simple

As noted above, including a complete model at part of the algorithm brings on new problems of model errors, model updates, measurement application, and so on. So application to a complete model is not included. Further, application of simple topology can be complicated, so while this is included, it is user optional and can be applied to the extent the user wishes. The algorithm is centered on the C37.118 flags which are well-known, established, and universally implemented.

2. C37.118 flags

These flags record the principal issues at the PMU and carry them through the entire measurement system. They are adapted to modification at each processing stage so can be used to indicate most problems as they are discovered.

3. Add more problem detection

Some effects are well known, such as the derivation of frequency from phase angle. Others, such as creation of noise on a signal, are not so clear. Stages have been added to flag many of these problems with user settable adjustments.

4. Imbed problem resolution in the data

Linking problems that are detected to the data itself simplifies use of the data. For this algorithm, all phasors are converted to floating point (FP) polar and frequency to FP real with scaling applied. Data may be received as integer or floating point in polar or rectangular coordinates. Conversion to one data type simplifies all further operations. Integer values converted to floating point (FP) do not lose any resolution and FP prevents over-range (overflow) problems. Scaling is applied, so all values can be used in engineering units. Polar coordinates separate phase angle and amplitude issues. Some data problems clearly affect only the phase angle, some magnitude only, and others both; polar representation allows differentiating which are affected. For example, timing problems affect only the phase angle and this can be indicated without affecting the amplitude.

FP representation includes a NaN value that clearly indicates that the value is not to be used. There is no equivalent for integer data. In this algorithm, when a value is unequivocally bad, it is set to NaN. In cases where it is questionable, the user can select to have it set to NaN or not. The user can receive data with bad and questionable data set to NaN so no flag comparisons are needed when using the data.

5. Add a data quality flag for each measurement

The C37.118 quality flags apply to a set of measurements from a PMU. They indicate conditions that apply to the whole data set from a particular PMU, though in many cases they only affect certain values.

For example, the sync error flag affects the phase angle measurement of all the phasors but not magnitude, frequency, or ROCOF. There also may be one phasor that is bad but the rest are good. The flags do not differentiate between measurements in the PMU measurement set. Once the measurements are received at a central site they need to be associated with the power system, so they need to stand alone with their own validation. This algorithm examines each measurement according to the relevant criteria and assigns an 8-bit flag to that measurement. The flag includes the applicable C37.118 criteria as well as the results of additional examination. These flags can be added to the data stream to indicate the validity for downstream applications.

6. Allow user to select data with flags to allow application determination of what to discard

A data output will be provided with all data as received except that it is converted to FP with scaling, phasors are in polar coordinates, unusable data is converted to NaN, and quality flags are added. As described above, floating polar simplifies processing and carries full resolution. It can be converted back to other formats if the user requires this conversion, but it is not recommended. Some data, such as filler for missing data, is always bad, and such data is always converted to NaN. NaN can be used for other data errors at the user's selection, but will not automatically be inserted. The quality flags are 8-bit and one is assigned to each variable. A phasor will have 2 flags, one for amplitude and one for phase angle. Frequency and ROCOF will each have one. Thus there will always be an even number of flags for a data frame. The flags will be packed into a 16-bit word and inserted into the data stream as an analog value in the order of the phasors and frequency/ROCOF that they apply to. The receiving application can thus use the flag to decide which data items are useful for a particular function.

7. Allow user to receive fully conditioned data where bad or uncertain data is set to NaN

This output is the same as above except the flags are not included in the data stream. The user can select which conditions will result in changing the data to NaN. In this case, it is anticipated the user will set data that is suspected bad to NaN where in the previous case they would let the data go through and that decision would be made by the application.

The next section details the algorithm that has been developed and its operation.

4. Development of the Data Validation and Conditioning Algorithm

A. The Overall Algorithm

The algorithm will be a process that examines the data and makes modifications as well as creating a quality flag for each synchrophasor data variable. Validation assures that data is uncorrupted, synchrophasor data. Invalid data is discarded. Validated data is passed to a series of comparison modules which condition the data by flagging as valid, invalid, or uncertain. Validated, conditioned data is combined with the flags in a C37.118 output stream. Applications use the validation and conditioning flags higher assurance data use.

Implementation is a series of sub-processes that examine different aspects of the data. The first three validate the data by examination of the packet contents and the communication that provides it. A data

packet may be declared valid and usable for further processing, or invalid and discarded. The next three sub-processes condition the data by looking at status flags and making comparisons among individual measurements and reasonability limits. Measurements that fail these tests will be set to invalid if they cannot be used with a high degree of confidence and uncertain if they cannot be reliably validated. In addition to conditioning, the algorithm will build a data Quality Flag (QF) for all validated data. This flag will indicate the type of error or uncertainty in the measurement if any is found. Output from the algorithm will be the original validated data without the quality flags or the conditioned data with the QF flags included as C37.118 analog values.

The algorithm will be designed as a real-time continuous flow process that adds minimal delay so the validated data can be used in real-time applications without latency problems. All valid data will be converted to polar floating point form with scaling applied. This simplifies processing by using actual engineering units without any loss of precision. It also allows separation of timing (phase angle) and magnitude problems as well as the use of not-a-number (NaN) indication for invalid measurements.

The algorithm implementation (Figure 2) will have a database of user entered parameters that set limits, decision points, topological comparisons, and other parameters that are needed to implement the algorithm. There will be a user interface (GUI) for entering these parameters. Since the input data configuration can change at any time, the algorithm will be designed so the comparisons follow the data items they are configured for. That is, a new channel may be added to an existing PMU or a new PMU may be added to the stream. The programmed comparisons need to follow the original elements for which they were configured. The algorithm will also have user settable defaults for any new or un-configured inputs. Similarly the examination and flags for data items that have been deleted from the data stream must be suspended. They may remain in the database for easy restoration if the signal is restored.

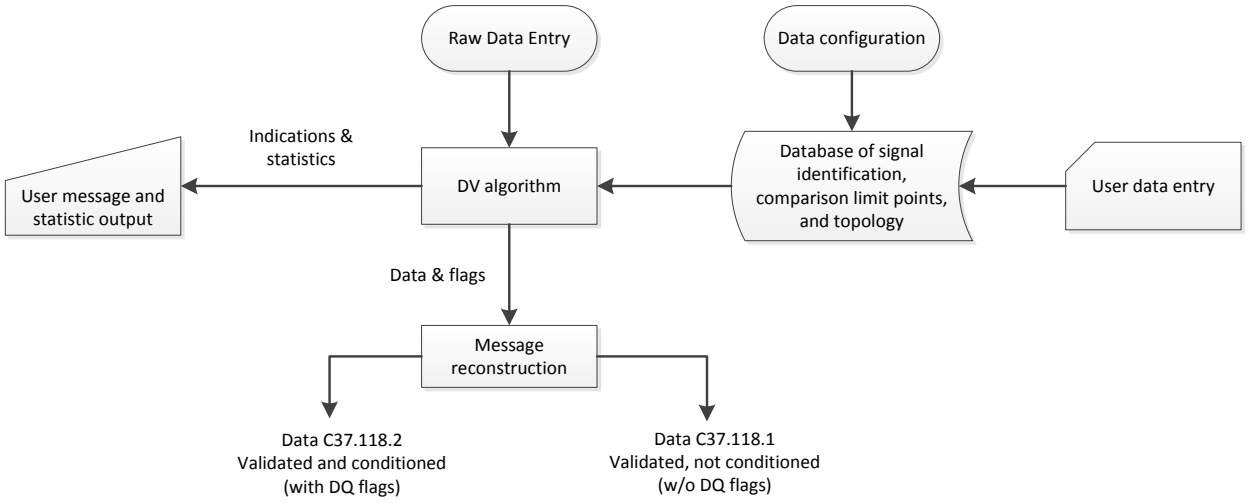

Figure 2. Overall algorithm process

The data input will be a real-time data stream in the C37.118 format (any version). Once processed by the algorithm, the algorithm will output the data to other applications in the C37.118.2 format. The

algorithm does not perform PDC functions; there is only 1 stream in and 1 stream out. It will handle streams with multiple PMUs. It will ask for config2 and config3 and use these for self-configuration.

The algorithm will process the data in successive modules as shown in Figure 3 below. Each module will examine particular characteristics of the data and record the outcome in an 8-bit quality flag. Each module should be independent of the others to permit simple maintenance and troubleshooting. The quality flag will be created for each measurement and carry through the process. The flag includes both bit indications and counts.

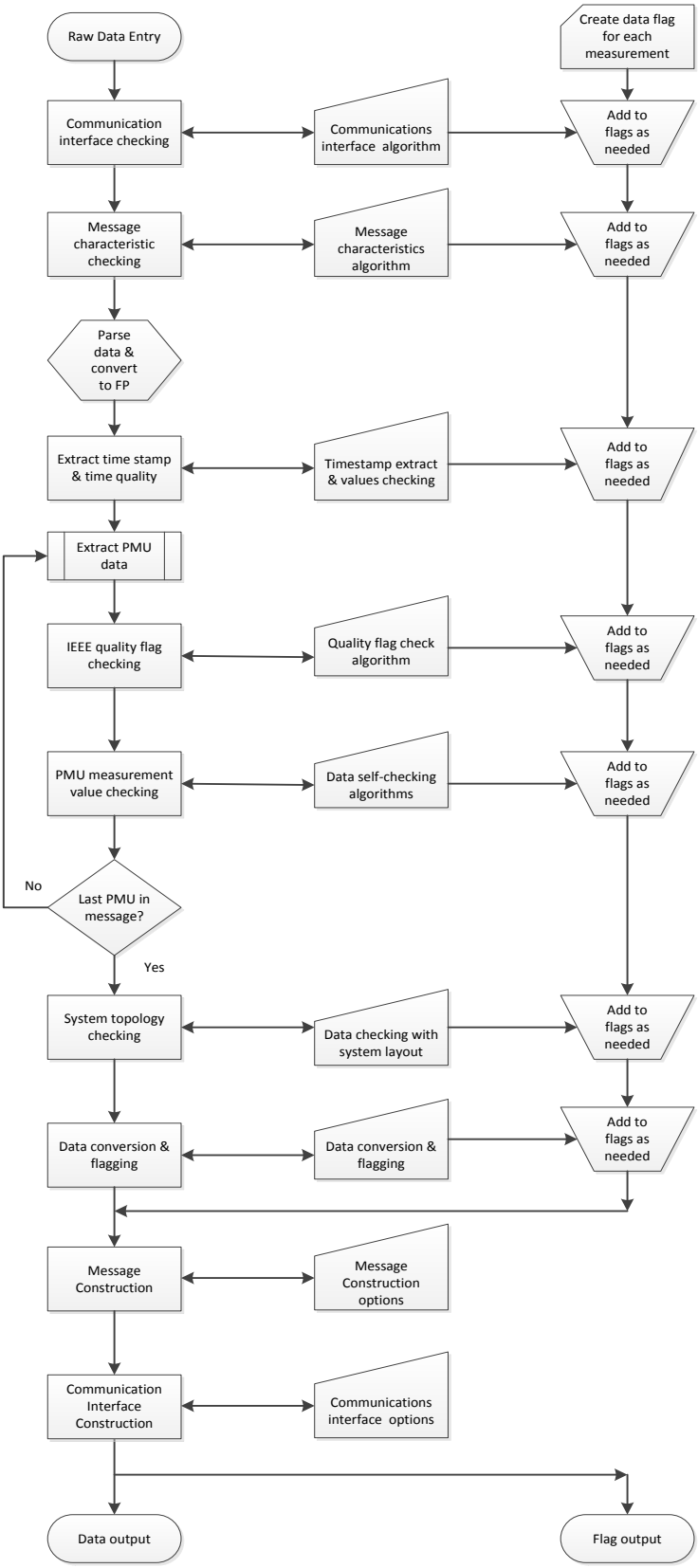

Figure 3. Algorithm module organization

In addition to the flag, there will be statistical and visual reporting. The algorithm will keep a record, including counts of each type of error or exception that includes the time detected. This will be in a sortable format so the user can compute statistics that can be used to track performance as well as aid in troubleshooting. It also will include a GUI display so the user can assess the system status at a glance.

B. Module 1 – Communication Interface

The first module (Figure 4) will process any information received from the communication interface. This will indicate errors such as framing and Ethernet CRC errors, if reported. Usually the interface will discard the data and no data will be forwarded. In that case, the error will be indicated as lost data.

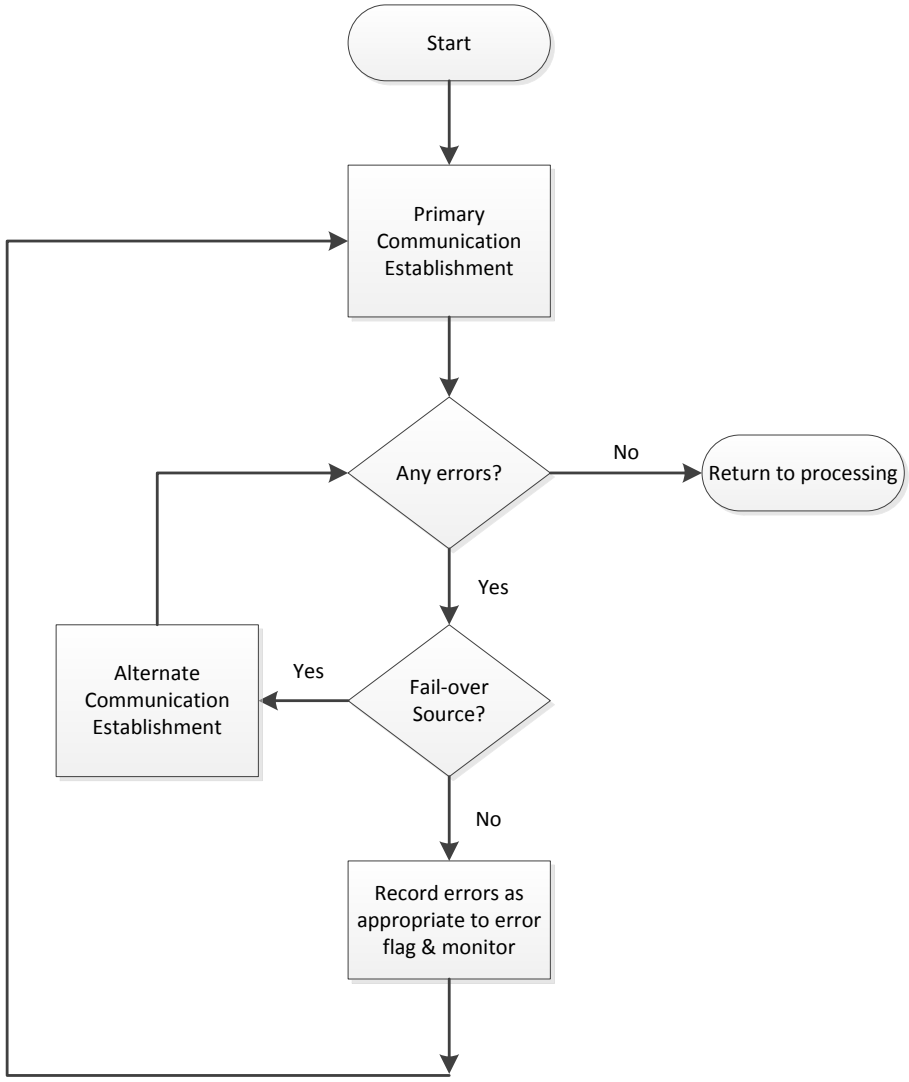

Figure 4. Module 1 – Communication Interface

C. Module 2 – Message Characteristics

The second module (Figure 5) will examine message characteristics such as message length, destination, etc. If there is a fatal error up to this point, the data is declared bad and no further checking is necessary. After the second module, the message needs to be parsed from input packet or stream. This is necessary since the individual data items will be examined separately. At this point the data should be intact enough that it can be reliably separated into message parameters and measured quantities.

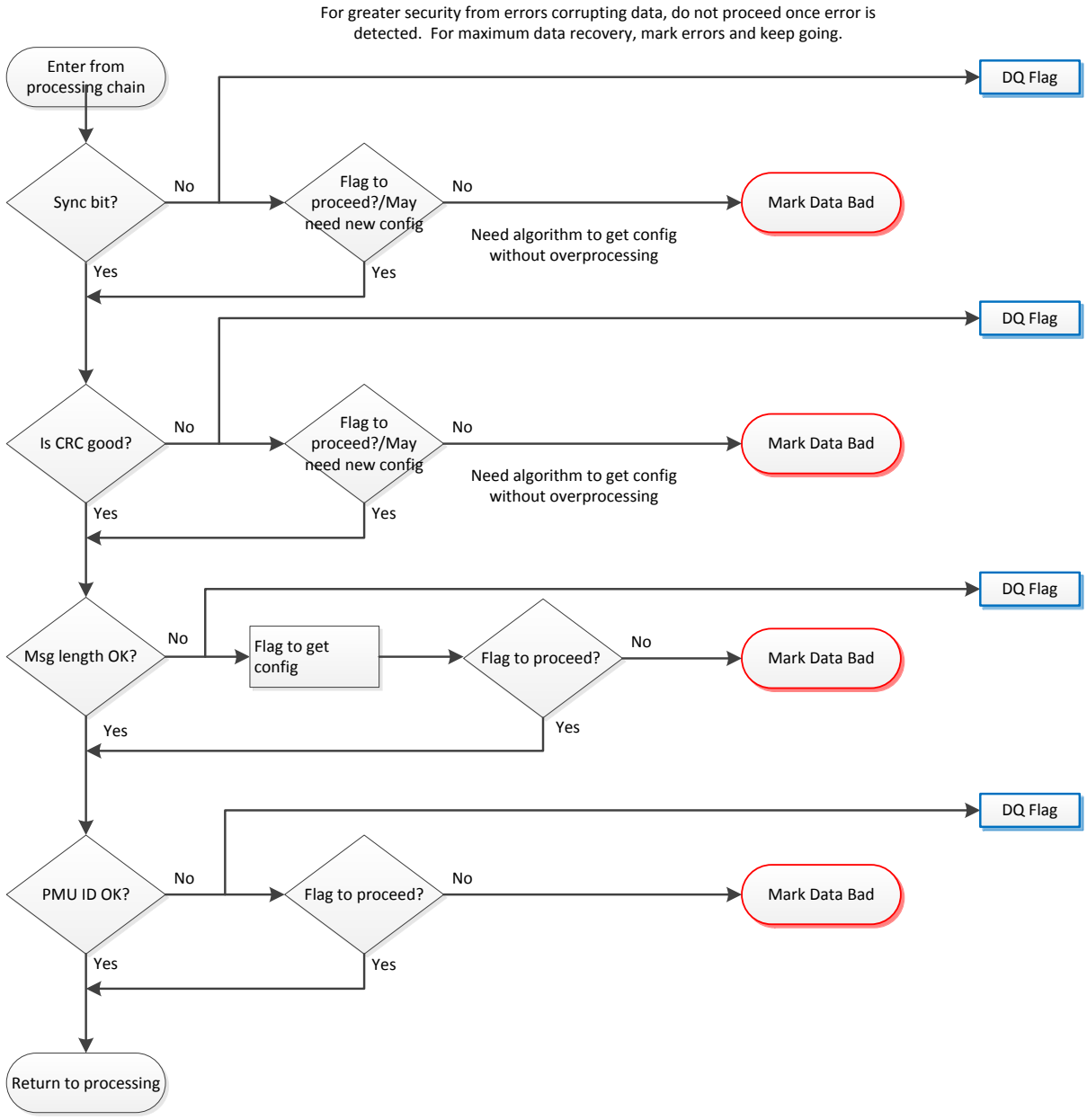

Figure 5. Module 2 – Message Characteristics

D. Module 3 – Timetag

The third module (Figure 6) extracts the timetag and examines it for errors. It checks to be sure the timetags are in sequence and none are missing based on the programmed data rate. If the timetags are in correct order it calculates latency based on arrival time. If accurate time is not supplied to the algorithm, this function can be skipped. It records any errors in the quality flag.

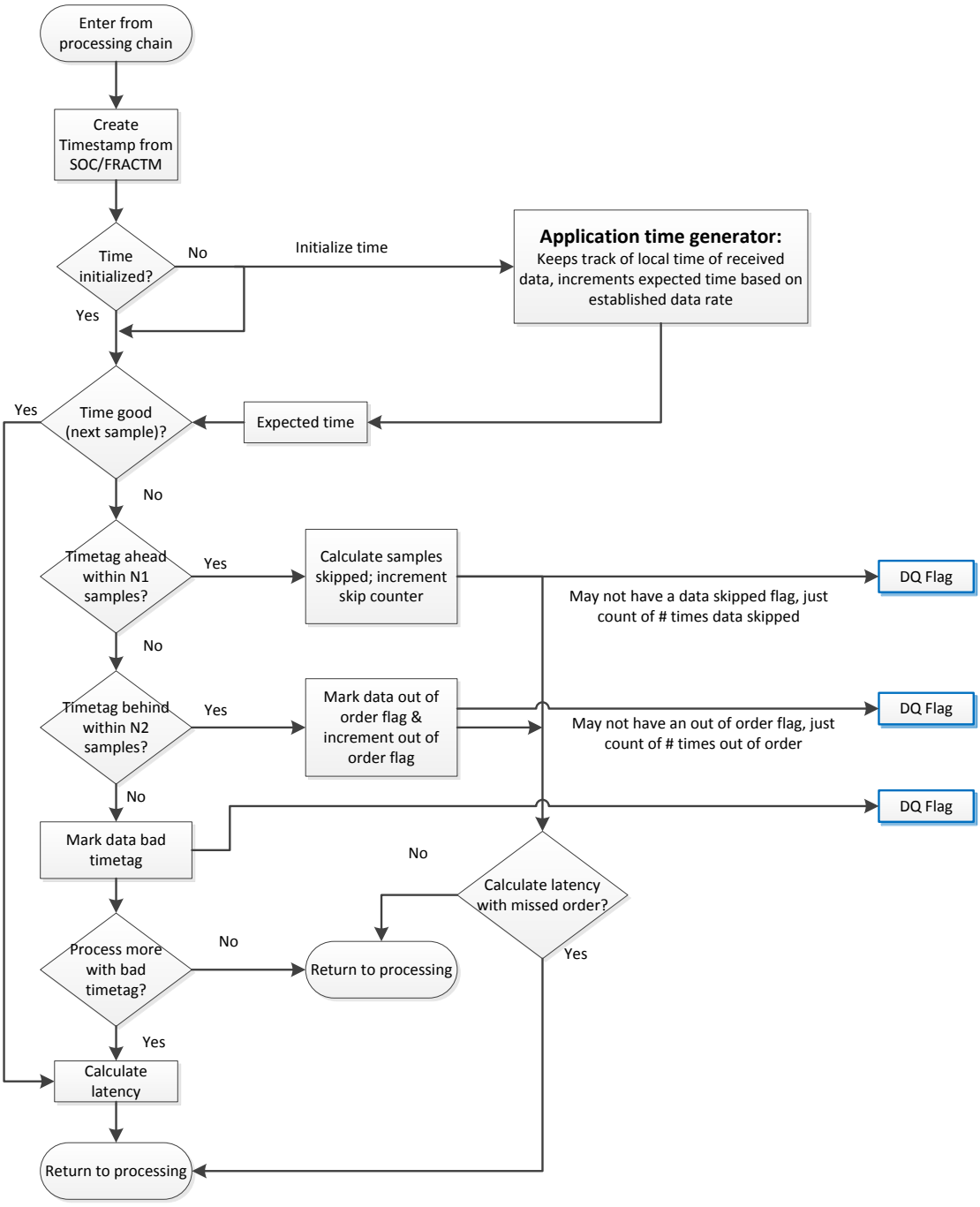

Figure 6. Module 3 – Timetag

E. Module 4 – Status Flag

The fourth module (Figure 7) looks at the C37.118 status flag for each individual PMU. Since these flags cover the whole PMU, which includes several measurements, these check results have to be applied to all the quality flags for each measurement from that PMU. This module checks for missing data, PMU errors, data out of sync, test mode data, and data with a local timetag. It applies criteria according to whether the data follows the 2005 or 2011 version indications. According to the user indications, it will mark the data bad, as well as report to the quality flag.

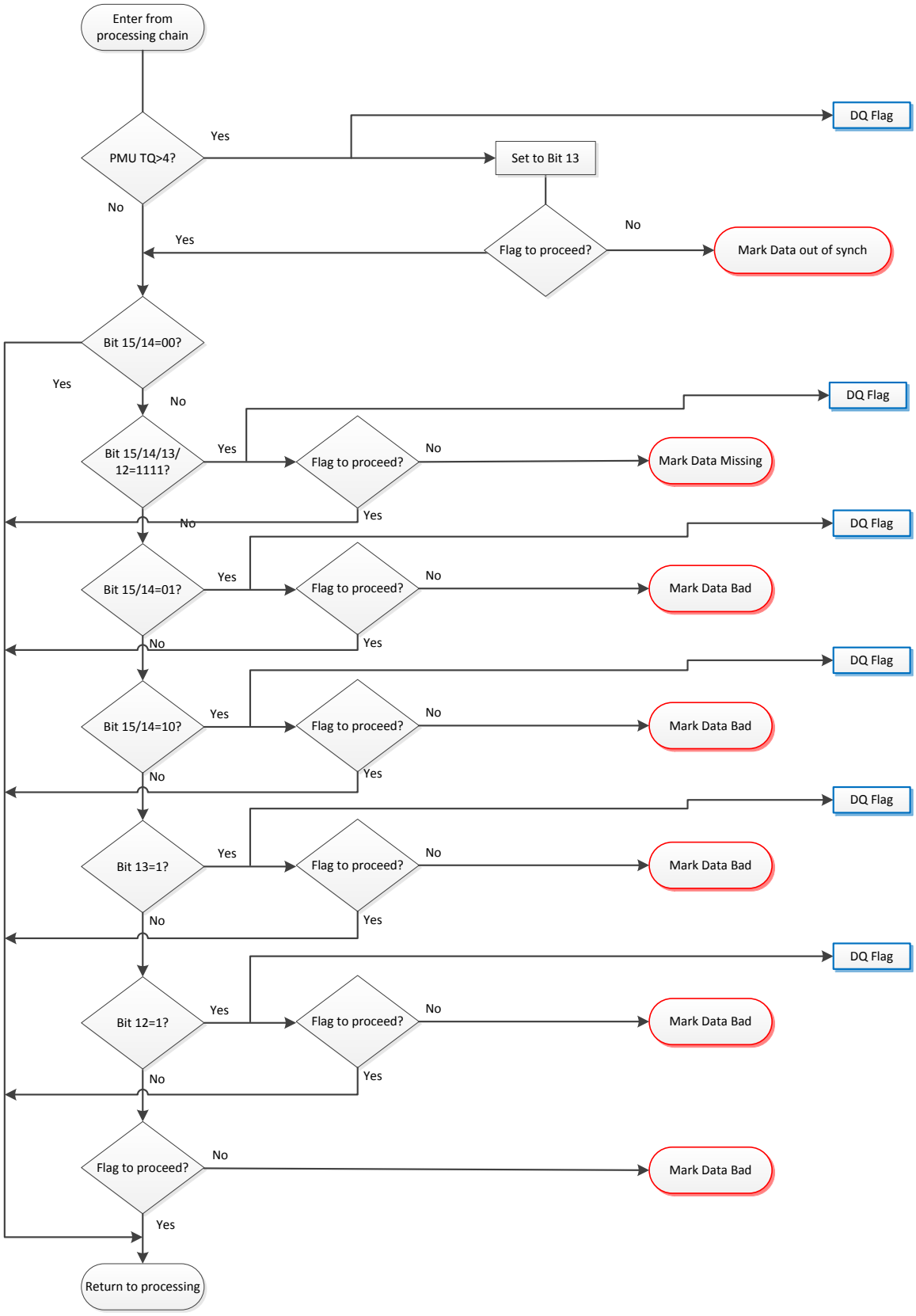

Figure 7. Module 4 – Status Flag

F. Module 5 – Data Characteristics

The fifth module (Figure 8) converts the data to floating point and applies scaling according to the configuration parameters. Floating point allows setting NaN uniquely which integer format does not. With floating point, all comparisons can be done relative to their normal values, simplifying the user interface. With polar, the error effects of timing and magnitude are separated and easier to test. After conversion, unusable invalid data can be changed to NaN. The user can specify that data declared suspect or invalid at certain points can also be changed to NaN. The point here is that if the user wants only "clean' data, all errors can be covered with NaN so status flag consideration is not necessary. This will often result in the disabling of otherwise usable data, so will not be the setting of choice in many cases. Data to be declared invalid in modules 1-4 will be set to NaN at this point.

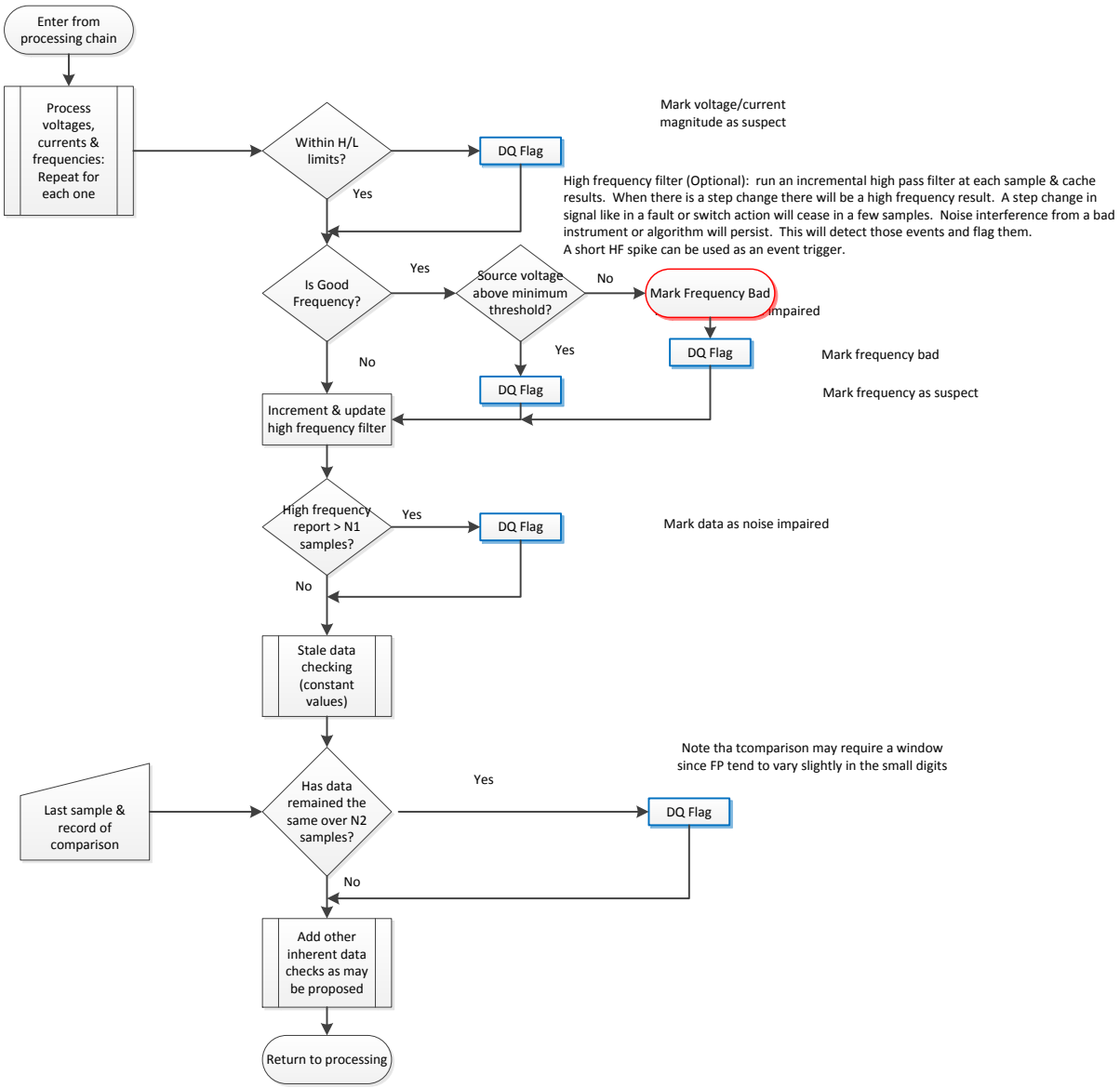

Figure 8. Module 5 - Data Characteristics

This module also incorporates self-checking algorithms. These include H/L limits for voltage, current, and frequency. Another detector invalidates the frequency if the signal that provides the measurement is absent. Random noise at frequencies higher than the normal passband (for the given reporting rate) indicates an interfering signal in most cases. A high frequency detector invalidates measurements that have significant interfering signals. Signals that do not change from sample to sample indicate a bad output; these are detected and flagged.

i. Range Check

Range check and stale detection will be performed for every enabled raw measurement which is marked as `GOOD` during PMU Status based detection.

Define:

M=signal value for voltage magnitude, current magnitude, or frequency

U=upper band

L=lower band

If a is infinite or Not a Number (NaN), *M* will be marked as `BAD` with sub-status as `Device Failure` and Limit Bit set to `High Limited`.

If (*M*>*U* OR *M*<L), *M* will be marked as `UNCERTAIN` with sub-status as `Engineering Unit Exceeded` and Limit Bit set to `Low Limited` or `High Limited`; Otherwise, mark data quality flag of a as `GOOD`.

The voltage angle carries the same quality flag as associated voltage magnitude. The current angle carries the same quality flag as associated current magnitude. Df/dt carries the same quality flag as associated frequency measurement.

RT-PDVC Management Tool provides user interface to configure ranges for every measurements.

ii. Stale Detection

Define:

M=signal value for voltage magnitude, current magnitude, or frequency

D=deviation limit

t= time where deviation exceeds limit

T= time window limit

If (max(*M*) – min(*M*)<=*D* AND *t*>=*T*), mark data quality flag of a as `UNCERTAIN` with sub-status as *Last Usable Value* and Limit Bit set to `Constant`. Otherwise, mark data quality flag of a as `GOOD`.

RT-PDVC Management Tool provides user interface to configure deviation limit and time window limit.

iii. Frequency Correlation Error Detection

The PMU may measure several voltages but calculate frequency and df/dt from one of them. That voltage can be used to validate PMU's frequency and df/dt. In this case, frequency and df/dt will carry the same quality flag as voltage.

iv. High Frequency Noise Detection

1. High Pass Filtering

Noise in the data that indicates an error could lead to an incorrect calculation or display. A legitimate oscillation should be detected and dealt with as a system problem. This is meant to detect noise that is not an oscillation.

Noisy detection will be applicable for voltage magnitude, current magnitude, and frequency measurements:

Define:

x = signal values for voltage magnitude, current magnitude, or frequency

n = total number of values

y = calculated value after applying high pass filter

data_rate = phasor measurement data rate

cutoff_frequency = the cutoff frequency of high pass filter

```
function high_pass(float[0..n] x, float data_rate, float cutoff_frequency)

   var float dt := 1 / data_rate
   var float RC := 1 / (2 * π * cutoff_frequency)
   var float α := RC / (RC + dt)
   var float[0..n] y

   y[0] := x[0]
   for i from 1 to n
     y[i] := α * (y[i-1] + x[i] - x[i-1])
   return y
```

2. Cutoff Frequency

The idea here is using a high-pass filter to detect activity that is higher than the normal range of oscillation but still within the range of reporting. If the samples were just randomly perturbed, it may appear as an oscillation at the Nyquist frequency. Thus there would be an oscillation at Fs/2 where Fs is the reporting rate. Since no legitimate frequency will be that high, the filter should be fully engaged at Fs/2. Typically with 30/s reporting, that would be 15 Hz. Legitimate oscillation has been seen at ~5 Hz. One should be able to exclude that frequency range. Therefore, the filter should have a response of approximately ½ full scale at 5 Hz, and >.9 at 15 Hz.

Figure 9. Data Rate 30, Cutoff Frequency 7.96 Hzshows the frequency response for data rate at 30 samples a second using 7.96 Hz as cutoff frequency. This comes close to the criteria, with half response (0.5) at 4.7 Hz and nearly full response (.88) at 15 Hz. The Cutoff frequency will be user configurable and default will be 7.96 Hz.

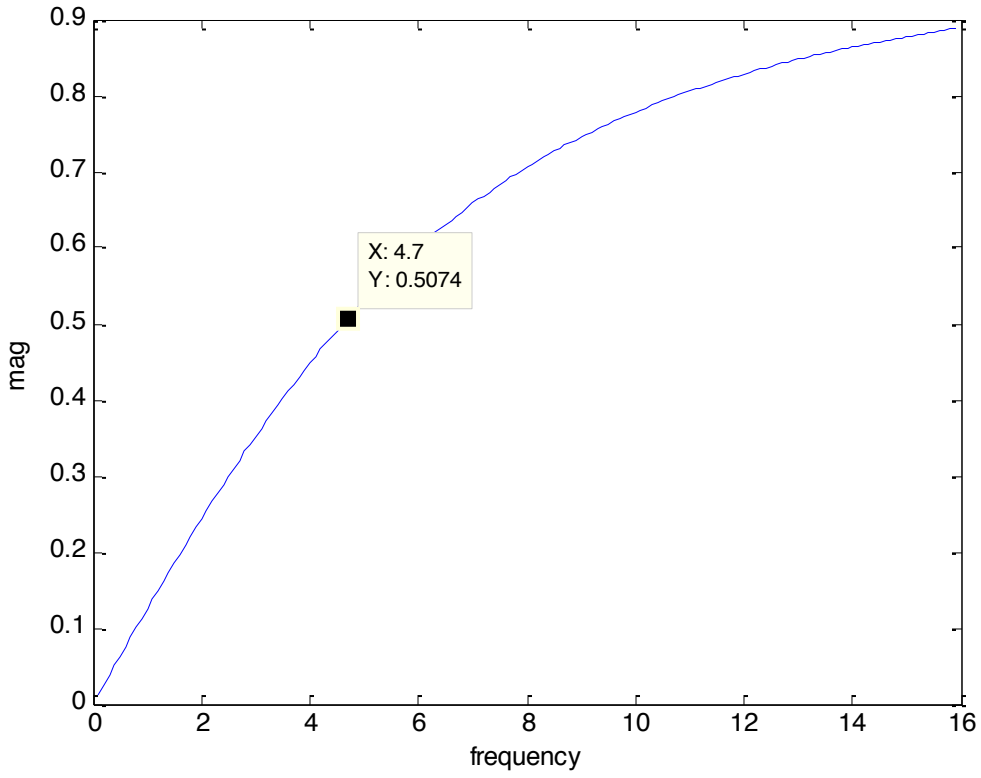

Figure 9. Data Rate 30, Cutoff Frequency 7.96 Hz

3. Noisy Detection

The standard deviation of high-pass filtered values will be calculated and checked against a user-defined limit (*T*). If (*y* >= *T*) the value is treated as noisy and its data will be marked as `NOISY`. Otherwise, the data is marked as `GOOD`.

G. Module 6 – Topology Comparisons

The sixth module (Figure 10) incorporates system topology checking. It performs data checks such as the sum of currents into a bus, voltages on the same bus, currents at two ends of a line, and similar type checks. It can include combining V & I for compute power and examining the limits for this. It will be implemented with user-defined equations including tolerance levels, so that checking and limits can be set according to the measurements with appropriate limits. The measurement comparisons relate to the actual system topology, so it digs deeper into the actual power system than the other quality checking. Also note, that in most cases, the test will involve several measurements and it will not be possible to positively conclude which measurement is at fault if the test indicates an error. So if the test fails, all measurements involved will be declared suspect; resolution will require further analysis.

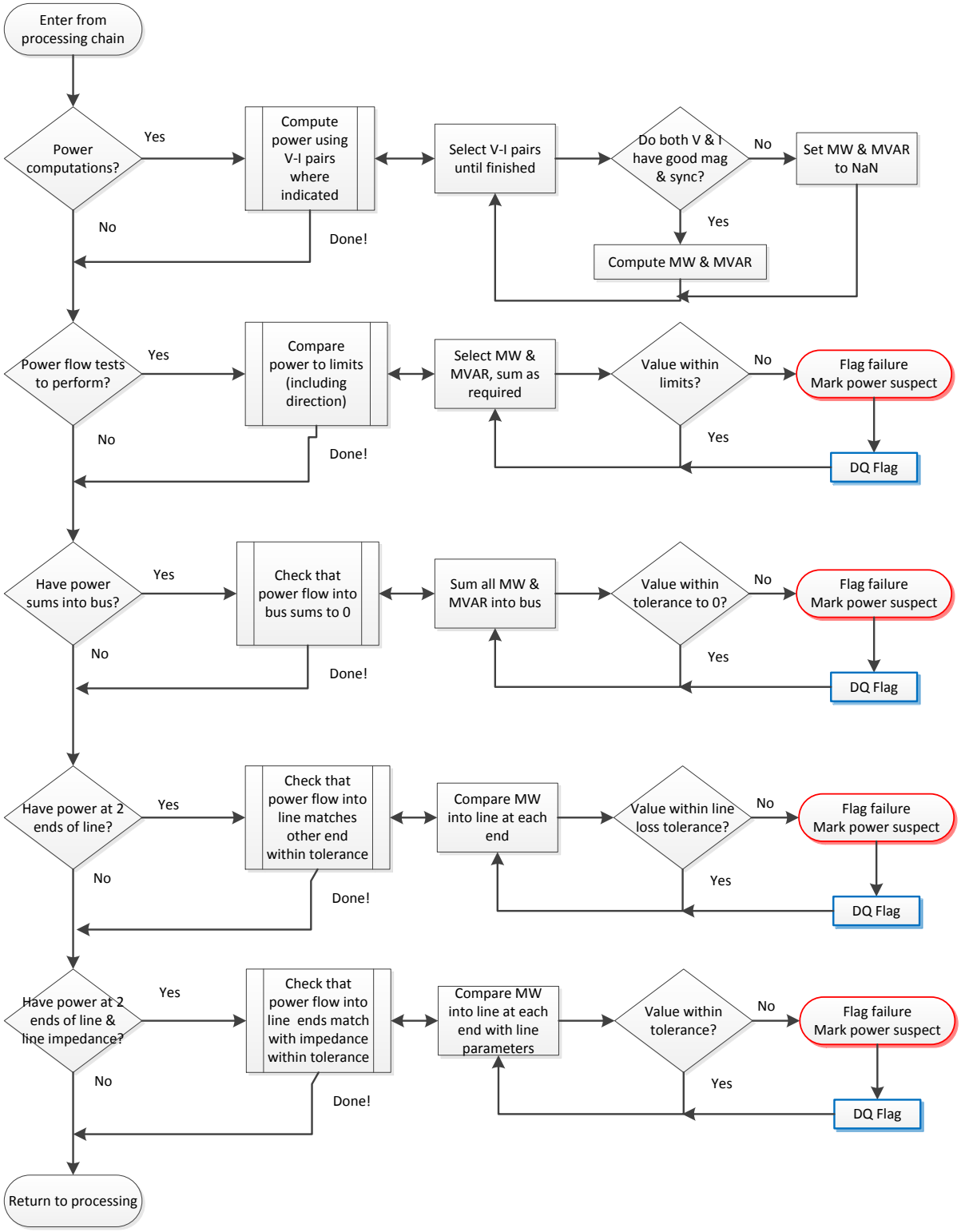

Figure 10. Module 6 – Topology Comparisons

For example, if the two PMUs measured the same line's voltage and current, the calculated power from one PMU measurement should match with the calculated power of another PMU. This example topology-based rule can be expressed as the following user defined calculation:

$$abs(V1m * I1m * \cos(V1a - I1a) - V2m * I2m * \cos(V2a - I2a)) * 0.001732 <= 0.00001$$

here

abs: function to get absolute value

V1m: Voltage 1 magnitude (kV), *V1a*: Voltage 1 angle (Radian)

I1m: Current 1 magnitude (Amp), *I1a*: Current 1 angle (Radian)

V2m: Voltage 2 magnitude (kV), *V2a*: Voltage 2 angle (Radian)

I2m: Current 2 magnitude (Amp), *I2a*: Current 2 angle (Radian)

0.001732: constant, convert to engineering unit of MW

<=: logic

0.00001: thread hold in MW

Real power (MW) and Reactive power (MVAR) are calculated as following:

$$M = V_m * I_m$$

$$P = V_a - I_a$$

$$MW = M * \cos(P) * 0.001732$$

$$MVAR = M * \sin(P) * 0.001732$$

Where

V_m: voltage magnitude (kV)

I_m: current magnitude (Amp)

V_a: voltage angle (Radian)

I_a: current angle (Radian)

H. Algorithm Outputs

Each module will add to the quality flag as appropriate. In some cases the test will need the result of a previous module. In some cases, the data will be set to invalid by a test. This should be indicated in the flag as well and can be used to discontinue further checking. Each module will also report test outcomes, usually in the form of error counts, such as the count of high voltages. The reporting component needs to totalize the counts on a periodic basis and save them, as well as display them, so the user can observe the activity. The module diagrams indicate where records are made in the quality flags, where data is marked bad, and parameters that the user needs to enter.

The data output will be user selectable (or all choices to different outputs). One output will be the "cleaned" data in the same form as received. This will be the same data with the exception that all phasors will be in floating point polar, frequency and ROCOF in floating, the status for each will be updated according to problems detected by the algorithm, and the bad values set to NaN. All analogs and digitals will be included as received. This output will include the DQ flags created by the algorithm. The second output will be the data as above except data found to be bad will be converted to NaN as indicated by the user. The user can indicate which of the questionable and suspect data will be invalidated and which will be left to be used.

4. Conclusion

This report details the analysis of the perceived problem with bad data from synchrophasor systems. The principal problems are fillers for lost data, un-validated measurements, and corrupted data. When these are used indiscriminately to produce displays, alarms, and analyses, overall confidence in the phasor data system is lost. These problems and many other more subtle problems are detectable. Once detected, bad or suspect data can be marked as "use with caution" or "do not use." To be effective, applications will need to be equipped to read and follow these warnings. An algorithmic process made up of 6 modules for data validation and conditioning to examine and flag data has been developed. A prototype implementation to demonstrate its use will be done in the next step as part of this project.

Appendix A. Mapping of specific problems to Modules where problem is addressed

The project statement of work listed specific problems that the algorithm should be able to detect. These are shown in column 1 of Table 1. The particular module of the algorithm where the problem is detected is indicated in column 2.

Table 1. Specific Problems and Module Mapping

Problem as listed in SOW	Module of Algorithm where detected
Loss of data from one or several PMUs	2. Message characteristics
Loss of signals in a PMU (assume loss of a signal input)	5. Data characteristics
Stale (non-refreshing) data	5. Data characteristics
Inconsistent data (don't understand this—assume HF noise)	5. Data characteristics
Inconsistent data rates - Inputs changing rates Different inputs at different rates	3. Timetag PDC function
Inconsistent latencies (assume latency changing over time)	3. Timetag
Off-sets in signal—Magnitude Phase angle	5. Data characteristics Manual phasor-EMS comparison
Corrupted and drifting signals in a PMU (This is nebulous, not well-defined. Some better defined conditions are given below.) Corrupted signals: rapidly changing, random values – high noise as above high error values – over-range values in data characteristics Drifting signals: Drifting magnitude – large drift, data characteristics - Small drift, not detected Drifting phase angle – when caused by a time error	 5. Data characteristics 5. Data characteristics 3. Timetag
Corrupted and drifting time reference in one or several PMUs If detected by the PMU, indicated by the status If not detected by PMU, it is not detected	4. Status flag
A combination of several issue described above	1-6. All modules work independently
Loss of primary source and/or communication and transition to alternate source (most system do not have a backup communication source. If equipped, the transfer mechanism will provide indication)	1. Communication interface